MATHEMATICS AND CYBERNETICS – APPLIED ASPECTS

Розроблено методику навчання штучних нейронних мереж для інтелектуальних систем підтримки прийняття рішень. Відмінна особливість запропонованої методики полягає в тому, що вона проводить навчання не тільки синаптичних ваг штучної нейронної мережі, але й виду та параметрів функції належності. В разі неможливості забезпечити задану якість функціонування штучних нейронних мереж за рахунок навчання параметрів штучної нейронної мережі відбувається навчання архітектури штучних нейронних мереж. Вибір архітектури, виду та параметрів функції належності відбувається з врахуванням обчислювальних ресурсів засобу та з врахуванням типу та кількості інформації, що надходить на вхід штучної нейронної мережі. За рахунок використання запропонованої методики не відбувається накопичення помилки навчання штучних нейронних мереж в результаті обробки інформації, що надходить на вхід штучних нейронних мереж. Також відмінною особливістю розробленої методики є те, що для обчислення даних не потрібні попередні розрахункові дані. Розробка запропонованої методики обумовлена необхідністю проведення навчання штучних нейронних мереж для інтелектуальних систем підтримки прийняття рішень, з метою обробки більшої кількості інформації, при однозначності рішень, що приймаються. За результатами дослідження встановлено, що зазначена методика навчання забезпечує в середньому на 10–18 % більшу високу ефективність навчання штучних нейронних мереж та не накопичує помилок в ході навчання. Зазначена методика дозволить проводити навчання штучних нейронних мереж за рахунок навчання параметрів та архітектури, визначити ефективні заходи для підвищення ефективності функціонування штучних нейронних мереж. Використання зазначеної методики дозволить зменшити використання обчислювальних ресурсів систем підтримки прийняття рішень та виробити заходи, що спрямовані на підвищення ефективності навчання штучних нейронних мереж; підвищити оперативність обробки інформації в штучних нейронних мережах

Ключові слова: штучні нейронні мережі, навчання, оперативність, обробка інформації, інтелектуальні системи підтримки прийняття рішень

UDC 681.324.01
DOI: 10.15587/1729-4061.2020.199469

DEVELOPMENT OF A METHODOLOGY FOR TRAINING ARTIFICIAL NEURAL NETWORKS FOR INTELLIGENT DECISION SUPPORT SYSTEMS

O. Sova
Doctor of Technical Sciences, Senior Researcher, Head of Department*
A. Shyshatskyi
PhD, Senior Researcher
Research Department of Electronic Warfare Development***
E-mail: ierikon12@gmail.com
Yu. Zhuravskyi
Doctor of Technical Sciences, Senior Researcher
Scientific Center
Zhytomyr Military Institute named after S. P. Koroliov
Myru ave., 22, Zhytomyr, Ukraine, 10004
O. Salnikova
Doctor of Public Administration, Senior Researcher,
Head of the Educational and Research Center
Educational and Research Center of Strategic Communications
in the sphere of National Security and Defense****
O. Zubov
PhD, Associate Professor
Department of Troop Control****
R. Zhyvotovskyi
PhD, Senior Researcher, Head of Research Department**
I. Romanenko
Doctor of Technical Sciences, Professor, Leading Researcher**
Ye. Kalashnikov
PhD, Head of Research Laboratory
Research Laboratory of Problems of Development
of Combat Use of Rocket Forces and Artillery****
A. Shulhin
PhD, Senior Researcher
Research Department
State Aviation Research Institute
Andryushchenko str., 6V, Kyiv, Ukraine, 01135
O. Simonenko
Senior Lecturer*
*Department of Automated Control Systems
Military Institute of Telecommunications and Informatization
named after Heroes of Kruty
Moskovska str., 45/1, Kyiv, Ukraine, 01011
**Research Department of the Development
of Anti-Aircraft Missile Systems and Complexes***
***Central Scientifically-Research Institute of Arming
and Military Equipment of the Armed Forces of Ukraine
Povitrofloskyi ave., 28B, Kyiv, Ukraine, 03049
****National Defense University of Ukraine named after Ivan Cherniakhovskyi
Povitrofloskyi ave., 28, Kyiv, Ukraine, 03049

Received date 20.02.2020
Accepted date 18.03.2020
Published date 30.04.2020

1. Introduction

Decision support systems (DSS) are actively used in all areas of human life. They received special distribution while processing large data sets, providing information support to the decision-making process by decision-makers.

Nowadays, artificial intelligence methods are the basis of existing DSS [1–10].

The creation of intelligent DSS was a natural continuation of the widespread use of classic DSS. Intelligent DSS provide information support for all production processes and services of enterprises (organizations, institutions), including product design, manufacturing and marketing, financial and economic analysis, planning, personnel management, marketing, support for product creation (operation, repair) and prospective planning. Also, these intelligent DSS have been widely used for specific military tasks, namely [1, 2]:

– planning of deployment, operation of communication systems and data transmission;

– automation of troops and weapons control;

– collecting, processing, and generalizing the intelligence about the status of intelligence objects and others.

The main tool for solving computational and other problems in modern intelligent DMSS are evolving artificial neural networks (ANN).

The prospect of using evolving ANN is due to the fact that ANN that do not have the ability to evolve do not meet the requirements for data processing efficiency and training capabilities. Evolving ANN have universal approximation properties and fuzzy inference capabilities, making them widely used for solving various problems of data mining, identification, emulation, forecasting, intelligent management, etc. They provide stable performance in conditions of nonlinearity, uncertainty, stochasticity and randomness, various kinds of disturbances and interferences.

Despite their successful application to address a wide range of data mining tasks, these systems have several drawbacks associated with their use.

The most significant disadvantages are as follows:

– complexity of system architecture choice. Generally, a model based on the principles of computational intelligence has a fixed architecture. In the context of ANN, the neural network has a fixed number of neurons and connections. In this regard, adapting the system to new data coming for processing that is different from the previous data may be problematic;

– batch and multi-epoch training require considerable time resources. Such systems are not adapted to operate online with a sufficiently high rate of new data to be processed;

– many of the existing systems of computational intelligence cannot determine the evolving rules by which the system develops, and can also represent the results of their work in terms of natural language.

Thus, the urgent task is to develop new training methods for ANN that will solve these difficulties.

2. Literature review and problem statement

In [3], an analysis of the properties of ANN, which were used to predict the concentration of air pollutants is carried out. The work emphasized that ANN have a low convergence rate and a local minimum. It is suggested to use an extreme training machine for ANN, which provides high efficiency of generalization at extremely high speed of training. The disadvantages of this approach include the accumulation of ANN errors during the calculations, the inability to select parameters and the type of membership function.

In [4], modeling of the adequacy of banking capital management in Ukraine is presented. The foregoing modeling is based on trend forecasting models. The multilayer perceptron is used for calculations. The training of this perceptron is limited only by the training of synaptic weights.

In [5], an operational approach for spatial analysis in the maritime industry is presented to quantify and display related ecosystem services. This approach covers the three-dimensionality of the marine environment, considering separately all marine areas (sea surface, water column and seabed). In fact, the method builds 3-dimensional models of the sea by estimating and displaying each of the three marine domains by adopting representative metrics. The disadvantages of this method include the impossibility of flexible adjustment (adaptation) of estimation models while adding (excluding) indicators and changing their parameters (compatibility and significance of indicators).

The work [6] presents a machine learning model for the automatic identification of requests and provision of information support services that are exchanged between members of the Internet community. This model is designed to handle a large number of messages from social network users. The disadvantages of this model are the lack of mechanisms to evaluate the adequacy of decisions made and high computational complexity.

In [7], the use of ANN for the detection of heart rhythm abnormalities and other heart diseases is presented. The error propagation algorithm is used as the ANN training method. The disadvantage of this approach is the limited learning of only synaptic weights, without learning the type and parameters of the membership function.

In [8], the use of ANN for avalanche detection is presented. The error propagation algorithm is used as the ANN training method. The disadvantage of this approach is the limited learning of only synaptic weights, without learning the type and parameters of the membership function.

The work [9] presents the use of ANN to identify problems of anomaly detection in home authorization systems. The «winner-takes-all» method is used as a method of training for the Kohonen's ANN. The disadvantages of this approach are the accumulation of errors in the learning process, the limited learning of only synaptic weights, without learning the type and parameters of the membership function, as well as the need to store previously calculated data.

In [10], the use of ANN for identifying anomaly detection problems in human encephalograms is presented. The method of ANN training is the method of fine-tuning of the ANN parameters. The disadvantages of this approach are the accumulation of errors in the learning process, the limited learning of only synaptic weights, without learning the type and parameters of the membership function.

In [12], the use of machine learning methods, namely ANN and genetic algorithms, is presented. A genetic algorithm is used as a method of ANN training. The disadvantage of this approach is the limited learning of only synaptic weights, without learning the type and parameters of the membership function.

In [13], the use of machine learning methods, namely ANN and differential search method, is presented. During the research, a hybrid method of ANN training is developed, based on the use of the algorithm of backpropagation and differential search. The disadvantage of this approach is the limited learning of only synaptic weights, without learning the type and parameters of the membership function.

In [14], the development of ANN training methods using a combined approximation of the response surface, which provides the least learning and forecasting errors, was carried out. The disadvantages of this method are the accumulation of errors during training and the inability to change the ANN architecture during training.

The work [15] describes the use of ANN to evaluate the performance of the unit using the previous time series of its performance. SBM (Stochastic Block Model) and DEA (Data Envelopment Analysis) models are used for ANN training. The disadvantages of this approach are the limitations in the choice of network architecture and training of only synaptic weights.

In [16], the use of ANN for the estimation of geomechanical properties is presented. The error propagation algorithm is used as the ANN training method. Improving the characteristics of the error propagation algorithm is achieved by increasing the training sample. The disadvantage of this approach is the limited learning of only synaptic weights without learning the type and parameters of the membership function.

The work [17] describes the use of ANN for the estimation of traffic intensity. The error propagation algorithm is used as the ANN training method. Improving the characteristics of the error backpropagation algorithm is achieved by using bandwidth connections between each layer so that each layer sets only a residual function over the results of the previous layer. The disadvantage of this approach is the limited learning of only synaptic weights without learning the type and parameters of the membership function.

The analysis of scientific works [1–17] showed that the well-known training methods are used to train artificial neural networks. These methods tend to focus on the training of synaptic weights or membership functions. The use of known algorithms (methods, techniques) for training artificial neural networks, even with improved characteristics, does not meet the requirements, namely:

– increasing the amount of information that can be processed by artificial neural networks;

– increasing the reliability of decision making by intelligent decision support systems;

– increasing the speed of adaptation of the architecture and parameters of artificial neural networks in accordance with the emerging tasks;

– prevention of deadlock situations during training artificial neural networks;

– ensuring the predictability of the learning process of artificial neural networks;

– ensuring the uniqueness of decisions made by intelligent decision support systems;

– ensuring that large data sets are computed in one epoch without storing previous calculations.

3. The aim and objectives of the study

The aim of the study is to develop a methodology for training artificial neural networks for intelligent decision support systems, which allows processing more information, with the uniqueness of decisions made.

To achieve this goal, the following objectives were set:

– to develop an algorithm for training artificial neural networks for intelligent decision support systems;

– to give an example of practical application of the proposed methodology.

4. Materials and methods of the research

The Kohonen network [2] refers to self-organizing networks. This means that they do not receive the desired output signal when the input learning vector is received, and as a result of the training, the network divides the input signals into classes, thus forming topological maps.

It should be mentioned that T. Kohonen's self-organizing map displays the input space of dimension n into the output space of dimension m.

The self-organizing map has a very simple architecture with direct information transfer. In addition to the zero (receptor) layer, it contains a single layer of neurons, which is very often called the Kohonen layer [2].

Let us take a closer look at the self-organizing map architecture. The n-dimensional input signal arrives at the network input. The network contains a single layer of m neurons that form a rectangular array on the plane.

Neurons are characterized by their location in the network. Each Kohonen layer neuron is connected to each input of the zero (input) layer by direct connections, as well as to all other neurons by cross connections.

Fig. 1 shows the 1D Kohonen's map.

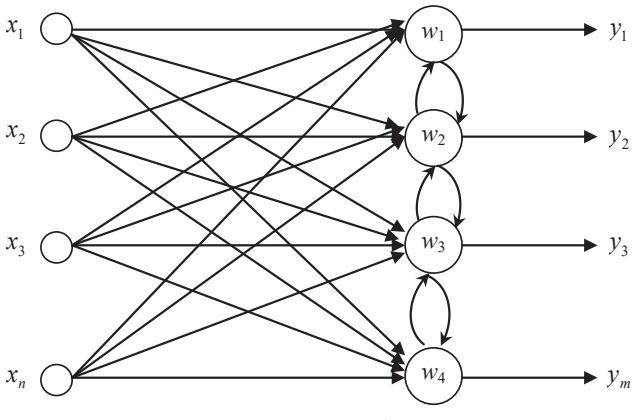

Fig. 1. 1D Kohonen's map

In the course of learning, neighboring neurons influence each other stronger than those located further away. Exactly lateral connections in the network provide excitation of some neurons and inhibition of others.

Each neuron from the Kohonen layer generates a weighted sum of signals:

$$f(x,w)=\sum_{i=1} w_i x_i.$$

If the synapses accelerate, then $w_{ij}>0$. If the synapses inhibit, then $w_{ij}<0$.

The classic Kohonen network training procedure is an adjustment of synaptic weights without taking into account other network learning options such as the type and parameters of the membership function and network architecture.

5. Development of an algorithm for training artificial neural networks for intelligent decision support systems

Fig. 2 shows the proposed algorithm for artificial neural network training. Improvement of the specified training algorithm consists in adding actions 3–8 to the known methods of training artificial neural networks.

There is additional training of artificial neural networks, which was not taken into account in [1–17]:

– learning the architecture of artificial neural networks, depending on the amount of input information (number of

layers, number of hidden layers, number of connections between neurons in a layer and between layers);

– learning the type and parameters of the membership function.

Step 1. The initial step is to initialize the initial values of synaptic weights.

Step 2. Determination of neuron weights.

Step 3. Correction of neuron weights and determination of neighborhood function.

Step 4. Formation of the first cluster.

Step 5. Verification of the threshold value.

Step 6. Checking the architecture's ability to handle the amount of information entering its input.

Step 7. Evolution of the system architecture.

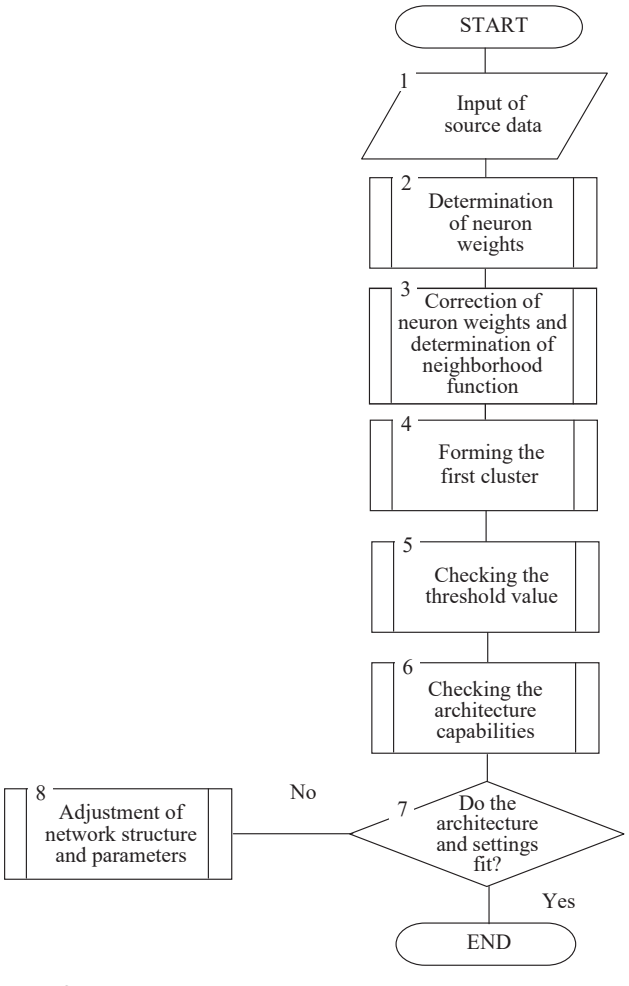

Fig. 2. Algorithm of functioning and training of an evolving artificial neural network

Before starting the Kohonen network learning algorithm, the input vectors are normalized:

$$\tilde{x}_i = \frac{x_i}{\sqrt{\sum_i x_i^2}} = \frac{x_i}{\|x\|}, \quad i = 1,2...,N. \tag{1}$$

The Kohonen network training algorithm itself can be described as a sequence of steps.

Step 1: initialize the initial values of synaptic weights $w_{ij} = 0$.

One commonly used initialization method is the assignment of values equal to randomly selected vectors from multiple observations to synaptic weights. In general, there are three main ways to initialize initial weights:

– initialization by random values, when all weights are given small random values;

– initialization by examples when the values of randomly selected examples from the training sample are given as initial values;

– linear initialization. In this case, the weights are initiated by the values of the linearly ordered vectors along a linear subspace passing between the two principal eigenvectors of the original data set.

Step 2: a normalized signal vector $\tilde{x}$ is fed to the system input and a vector of weights (neurons) closest to $\tilde{x}$, that is the vector, for which the Euclidean distance to $\tilde{x}$ is the smallest, is selected:

$$\arg\min_j \left\| \tilde{x} - w_j \right\|, \quad j = 1,2,...,l. \tag{2}$$

This expression can also be written in the following form:

$$A = \arg\max \left(\tilde{x}^T w_j \right), \quad j = 1,2,...,l. \tag{3}$$

Step 3: correction (adjustment) of the vector of synaptic weights w_{ij} according to the rule:

$$w_{ij}(k+1) = w_{ij}(k) + \eta(k) f_{ij}(k) \left(\tilde{x}(k) - w_{ij}(k) \right), \tag{4}$$

where $w_{ij}(k+1)$ is the new weight value; $\eta(k)$ is the gain coefficient changing over time (at the first iteration $\eta = 1$ gradually decreases to zero, that is, $\eta(k) \in (0,1]$; $f_{ij}(k)$ is a monotonically decreasing function (neighborhood function) of the type,

$$f_{ij}(k) = f\left(\left\| r_i - r_j \right\|, k \right), \tag{5}$$

where r_i is the vector that determines the position of the i-th neuron in the lattice, r_j is the vector that determines the position of the j-th neuron in the lattice.

Obviously, for the neuron-winner:

$$f_c\left(\left\| r_i - r_j \right\| \right) = f_c(0) = 1.$$

Most often, the Gaussian is chosen as a neighborhood option. In this case, the network output signals are defined as follows:

$$y_j(k) = \begin{cases} 1, \text{ if } w_j^T(k)\tilde{x}(k) > w_p^T(k)\tilde{x}(k) \text{ for all } p \neq j, \\ 0, \text{ otherwise.} \end{cases} \tag{6}$$

Then, steps 1 and 2 are repeated until the initial network values are stabilized with the specified accuracy. The meaning of tuning by expression (4) is to minimize the difference between the input vector $\tilde{x}(k)$ and the vector of the synaptic weights of the neuron-winner.

In other words, in the process of tuning, this algorithm «brings» the vector of synaptic weights of the neuron-winner to the current input image $\tilde{x}(k)$. In this case, there is a vector $\tilde{x}(k) - w_{ij}(k)$, which is then reduced by a value $\eta(k)$ that sets the learning speed. Thus, it can be said that training is reduced to rotating the vector of the neuron weights in the direction of the input vector without significantly changing its length.

The work [18] proposed an original online evolving fuzzy clustering method (EFCM) based on a probabilistic approach to solve a problem. The main parameter that ultimately

determines the final result is the radius of the formed clusters, selected from empirical considerations and ultimately determines the number of possible classes. Despite the effectiveness of fuzzy clustering (FC) probabilistic algorithms, their «weakness» is the «hard» restriction:

$$\sum_{j=1}^{m} u_j(k) = 1 \forall k = 1, 2, ..., N. \quad (7)$$

This disadvantage is not present in the so-called (possibilistic) fuzzy clustering approach (PCM) [17] associated with the optimization of the objective function. However, the observation also belongs to all classes, that is, is equidistant from all centroids, but does not belong to any of the clusters.

$$E(u_j, c_j) = \sum_{k=1}^{N}\sum_{j=1}^{m} u_j^{\beta}(k)\left\|x(k) - c_j\right\|^2 + \sum_{j=1}^{m}\mu_j\sum_{k=1}^{N}\left(1 - u_j(k)\right)^{\beta}, \quad (8)$$

where $c_j = \left(c_{j1}, c_{j2}, ..., c_{jn}\right)^T$ is the center of gravity of the j-th cluster, which is calculated during data processing; $\beta > 1$ is the fuzzyfication parameter («fuzzifier»), which determines the «blurring» of boundaries between clusters and is usually relied upon $\beta = 2$; $\mu_j > 1$ is the scalar parameter that determines the distance at which the membership takes a value of 0.5, if

$$\left\|x(k) - c_j\right\|^2 = \mu_j, \quad (9)$$

then $\mu_j(k) = 0.5$.

Using the possibilistic approach leads to the evolving fuzzy clustering method (EFCM), which is conveniently written in the form of a sequence of steps.

Step 4: upon receipt of the $x(1)$ observation, the first cluster with the c_1 centroid is formed.

Step 5: upon receipt of the $x(1)$ observation, the following condition is checked:

$$\left\|x(2) - c_1\right\| \le \Delta, \quad (10)$$

where the threshold Δ is set a priori.

If this condition is fulfilled, then the $x(1)$ observation does not form a new center of gravity, meaning that it must belong to the first cluster with the membership level:

$$u_1(2) = \left(1 + \left(\frac{\left\|x(2) - c_1\right\|^2}{\mu_1}\right)^{\frac{1}{\beta-1}}\right)^{-1}. \quad (11)$$

If the condition is fulfilled:

$$\Delta < \left\|x(2) - c_1\right\| \le 2\Delta, \quad (12)$$

then the centroid is corrected according to the WTA (winner-takes-all) Kohonen self-learning rule [18]:

$$c_1(2) = c_1(1) + \eta(2)\left(x(2) - c_1(1)\right), \quad (13)$$

$\eta(2)$ is the step setting parameter.

The centroid c_1 is brought to the $x(1)$ observation vector; if for $x(1)$, the following inequality holds:

$$2\Delta < \left\|x(2) - c_1\right\|, \quad (14)$$

then a second cluster with a centroid is formed:

$$c_2 = x(2). \quad (15)$$

In this case, the membership levels $u_2(1)$ and $u_1(2)$ must be calculated by the formulas below.

Step 6: Checking the architecture's ability to handle the amount of information entering its input.

Therefore, if there are N observations and m clusters with c_j centroids, the calculations of all the members and the adjusted coordinates of the centroids are estimated according to the relation:

$$\begin{cases} u_j(k) = \left(1 + \left(\dfrac{\left\|x(k) - c\right\|^2}{\mu_j}\right)^{\frac{1}{\beta-1}}\right)^{-1}, \\ c_j = \dfrac{\sum\limits_{k=1}^{N} u_j^{\beta}(k)x(k)}{\sum\limits_{k=1}^{N} u_j^{\beta}(k)}, \\ \mu_j(k) = \dfrac{\sum\limits_{k=1}^{N} u_j^{\beta}(k)\left\|x(k) - c_j\right\|^2}{\sum\limits_{k=1}^{N} u_j^{\beta}(k)}, \end{cases} \quad (16)$$

we obtain as a result of minimization (8) for all evaluated parameters.

The system of equations (16) is essentially a batch algorithm for processing information so that upon receipt of the observation $x(N+1)$, all calculations must be performed again. It is clear that at a sufficiently high data rate, the approach may be ineffective.

For this purpose, it is necessary to develop recurrent procedures that do not require storing previously processed data. These recurrent procedures can be implemented on the basis of a two-layer adaptive neuro-fuzzy network with the following architecture.

The first hidden layer of the network is formed by the ordinary Kohonen neurons N_j^K, connected by lateral connections through which the competition process takes place. The output layer of the network, formed by the nodes N_j^u, is designed to calculate the levels of membership $x(k)$ of each observation in each j-th cluster, $j = 1, 2, 3, ..., m$. To set up the cluster centroids, a recurrent self-learning procedure is used, which has the form [10]:

$$\begin{cases} c_j(k+1) = c_j(k) + \dfrac{u_j^{\beta}(k)}{k+1}\left(x(k+1) - c_j(k)\right), \\ u_j(k+1) = \dfrac{1}{1 + \left(\dfrac{\left\|x(k+1) - c_j(k+1)\right\|^2}{\mu_j(k)}\right)^{\frac{1}{1-\beta}}}, \\ \mu_j(k+1) = \dfrac{\sum\limits_{p=1}^{k+1} u_j^{\beta}(p)\left\|x(p) - c_j(k+1)\right\|^2}{\sum\limits_{p=1}^{k+1} u_j^{\beta}(p)}. \end{cases} \quad (17)$$

It is easy to see that the first expression (17) is a WTM self-learning rule with a narrowing neighborhood function $(k+1)^{-1}u_j^{\beta}(k)$.

Step 7. Evolution of the system architecture.

The process of system evolution, like the previous one, begins with a single Kohonen neuron that specifies the coordinates of the first c_1 centroid. The following neuron is added to the network when the condition (14) holds, which in this case takes the form:

$$2\Delta < \left\| x(k) - c_1(k-1) \right\|. \quad (18)$$

The neuron with a centroid $c_2(k) = x(k)$ is formed. It should be mentioned that since the data is pre-normalized to the hyperspace in the Kohonen neural networks, then

$$\left\| x(k) \right\|^2 = \left\| c_j(k) \right\|^2 = 1, \quad (19)$$

inequality (14), which determines the need for the introduction of new neurons into the network, has a form:

$$-1 \le 1 - 2\Delta^2 < c_j^T(k-1)x(k) \le 1, \quad \forall j = 1,2,...,m, \quad (20)$$

or

$$-1 \le 1 - 2\Delta^2 < \cos\left(c_j(k-1)x(k)\right) \le 1. \quad (21)$$

Thus, the build-up of the architecture occurs as a result of the constant control of inequalities (29) or (21), which occurs if these inequalities are violated.

Using the possibilistic approach, it is advisable to implement another «branch of evolution», namely, if at some point in time it turns out that the observation membership $x(k)$ do not exceed some threshold value:

$$u_j(k) < \varepsilon \forall j = 1,2,...,m, \quad (22)$$

The membership threshold of the observation $x(k)$ is determined on the basis of the computing capabilities of the hardware platform on which the decision support system is based.

The observation $x(k)$ is far enough away from all the already formed centroids. This can also indicate the formation of a new cluster:

$$c_{m+1}(k) = x(k). \quad (23)$$

The popular Xie-Beni index [10] can be used to estimate the quality of fuzzy clustering [14]. For a fixed sample with N observations, the index has the form:

$$EXB(N) = \frac{\left(\sum_{k=1}^{N} \sum_{j=1}^{m} u_j^{\beta}(k) \left\| x(k) - c_j(N) \right\|^2 \Big/ N \right)}{\min\limits_{j \ne l} \left\| c_j(N) - c_l(N) \right\|^2} = \frac{NEXB(N)}{DEXB(N)}. \quad (24)$$

For online processing, this index, like centroid clusters, can also be calculated recurrently:

$$EXB(k+1) = \frac{NEXB(k+1)}{DEXB(k+1)} = \frac{NEXB(k) + \frac{1}{k+1}\left(\sum_{j=1}^{m} u_j^{\beta}(k+1) \left\| x(k+1) - c_j(k+1) \right\|^2 - NEXB(k) \right)}{\min\limits_{j \ne l} \left\| c_j(k+1) - c_l(k+1) \right\|^2}. \quad (25)$$

Adding the expression (23) to the learning procedure (17) will allow additional control over the number of clusters formed by the system. Thus, entering the third threshold δ and checking the condition:

$$EXB(k+1) > \delta, \quad (26)$$

at every step make it is possible to stop the process of neurons build-up in the event of a violation of inequality (26).

6. Example of the practical use of the proposed method

The method of training artificial neural networks for intelligent decision support systems is proposed. The proposed methodology is simulated in the MathCad 14 software environment.

A two-dimensional artificially generated dataset was used for the experiment. It contained 15 clusters with different levels of overlap. The data sample contained 5,000 observations. Data were submitted for sequential processing.

Consider the example of setting up cluster centers for the evolving Kohonen neural network to solve clustering problems.

In the first stage of setting up centers, there are three situations:

1. For the distance between the new observation and the centers of the existing clusters, the condition is satisfied:

$$\exists_j \ \Delta < \left\| x(k) - c_j \right\| \le 2\Delta. \quad (27)$$

In this case, the clusters mix their centers towards the new observation. It should be mentioned that the movement of the cluster center depends on the parameter $\eta(k)$.

When a new observation arrives, the cluster center shifts toward that observation. In this example, the parameter $\eta(k) = 0.5$.

2. For the distance between the new observation and the centers of the existing clusters, the condition is satisfied:

$$\forall_j \ 2\Delta < \left\| x(k) - c_j \right\|. \quad (28)$$

In this case, a new cluster is created. The new observation becomes the center of the new cluster.

3. For the distance between the new observation and the centers of the existing clusters, the condition is satisfied:

$$\forall_j \ \left\| x(k) - c_j \right\| \le \Delta. \quad (29)$$

In this case, this observation belongs to existing clusters, and the centers of these clusters do not change.

To compare the quality of clustering, FCM and a system based on the evolving fuzzy clustering method (EFCM) with different threshold parameter values were used.

The Xie-Beni index was used as the criterion for assessing the quality of clustering.

Table 1 presents the comparative results of clustering.

For the next experiment, a data sample describing the characteristics of the monitoring object was used (Table 2). Each observation was described by seven parameters:

– number of personnel;

– total number of personnel;

– number of organizational and staff structures;
– number of samples of weapons and military equipment;
– number of communication devices;
– number of types of armaments and military equipment and types of communication.

Before clustering, the features of the observations were normalized at the interval [0, 1].

Table 1

Comparative results of clustering

System	Number of clusters	Algorithm parameters	XB (Xie-Beni index)	Clustering time, s
FCM (Fuzzy C-Means)	15	–	0.1903	2.69
EFCM	9	Dthr=0.24	0.1136	0.14
EFCM	12	Dthr=0.19	0.1548	0.19
Proposed system (batch mode)	12	delta=0.1	0.0978	0.37
Proposed system (online mode)	12	delta=0.1	0.1127	0.25

Table 2

Comparative results of clustering

System	Number of clusters	Algorithm parameters	XB (Xie-Beni index)	Clustering time, s
FCM (Fuzzy C-Means)	3	Dthr=0.6	0.2963	0.81
EFCM	4	Dthr=0.6	0.2330	0.54
Proposed system (batch mode)	3	delta=0.4	0.2078	0.45
Proposed system (online mode)	3	delta=0.4	0.2200	0.30

The parameters of the analyzed systems, as well as the number of identified clusters, are also given in Table 2. The Xie-Beni (XB) index was used to evaluate the quality of the systems. It should be mentioned that the proposed system showed a better PC (partition coefficient) result than the EFCM and a better running time result than the FCM. The proposed systems and FCM have identified three fuzzy clusters.

The research of the developed methodology showed that the mentioned training method provides on average 10–18 % higher efficiency of training of artificial neural networks and does not accumulate errors during training (Tables 1, 2)

These results can be seen from the results in the last lines of Table 1, 2, as the difference of the Xie-Beni index.

7. Discussion of the results on the development of methods for training artificial neural networks

The advantages of this method are achieved by performing a series of additional procedures, namely 3–8, which are shown in Fig. 2.

The analytical dependencies for performing these procedures are given in the formulas, namely:

– dependencies (4)–(9) – step 3 «Correction of neuron weights and determination of neighborhood function»;

– upon receipt of the observation $x(1)$, the first cluster with the c_1 centroid is formed;

– dependencies (10)–(15) – step 5 «Checking the threshold value»;

– dependencies (16), (17) – step 6 «Checking the architecture capabilities»;

– dependencies (18)–(26) – step 7 «Checking the architecture capabilities and parameters» (Tables 1, 2).

Thus, not only synaptic weights, but also other parameters of artificial neural networks are adjusted, which provides more accurate tuning of artificial neural networks.

The main advantages of the proposed evaluation methodology are:

– processing more information by reducing the improved learning algorithm;

– increased reliability of the obtained results of the absence of learning error accumulation during training artificial neural networks. This is achieved by adjusting the parameters and architecture of the artificial neural network;

– wide scope (decision support systems);

– simplicity of mathematical calculations;

– prevents the phenomenon of «retraining» by adjusting the network architecture;

– no need to store the results of previous calculations.

The disadvantages of the proposed methodology include:

– loss of information in the estimation (forecasting) due to the construction of the membership function. This loss of information can be reduced by choosing the type of membership function and its parameters in the practical implementation of the proposed methodology in decision support systems. The choice of the membership function depends on the computing resources of a particular electronic computing device.

– lower accuracy of estimation on a separate parameter of state estimation;

– loss of accuracy of results during the reorganization of the artificial neural network architecture.

The method will allow:

– training artificial neural networks;

– identifying effective measures to improve the efficiency of artificial neural networks;

– increasing the efficiency of artificial neural networks by learning the parameters and architecture of networks;

– reducing the use of computing resources of decision support systems;

– developing measures aimed at improving the efficiency of training artificial neural networks;

– increasing the efficiency of information processing in artificial neural networks.

The areas of further research should be aimed at reducing the computational cost of processing different data types in special-purpose systems.

8. Conclusions

1. The algorithm of training artificial neural networks for intelligent decision support systems is developed.

Improving the efficiency of information processing and reducing the estimation error are achieved by:

– training not only of the synaptic weights of the artificial neural network, but also the type and parameters of the membership function;

– training the architecture of artificial neural networks;

– calculating data for one epoch without having to store previous calculations. This reduces the time of informa-

tion processing due to the absence of the need to access the database;

– no accumulation of errors of training artificial neural networks as a result of processing the information coming to the input of artificial neural networks.

2. An example of using the proposed methodology on the example of clustering of the monitoring object is given. This example showed an increase in the efficiency of artificial neural networks at the level of 10–18 % by the Xie-Beni index and efficiency of information processing through the use of additional training procedures for artificial neural networks.

Acknowledgments

The author team is grateful for the help in the preparation of the paper:

To the doctor of technical sciences, professor Oleksiy Viktotovich Kuvshinov – deputy head of the Ivan Chernyakhovsky national defense university of Ukraine.

To the candidate of technical sciences, associate professor Oleksandr Mykolayovich Bashkirov – leading researcher at the central armed forces research institute of the armed forces of Ukraine.

References

1. Kalantaievska, S., Pievtsov, H., Kuvshynov, O., Shyshatskyi, A., Yarosh, S., Gatsenko, S. et. al. (2018). Method of integral estimation of channel state in the multiantenna radio communication systems. Eastern-European Journal of Enterprise Technologies, 5 (9 (95)), 60–76. doi: https://doi.org/10.15587/1729-4061.2018.144085
2. Kuchuk, N., Mohammed, A. S., Shyshatskyi, A., Nalapko, O. (2019). The method of improving the efficiency of routes selection in networks of connection with the possibility of self-organization. International Journal of Advanced Trends in Computer Science and Engineering, 8 (1), 1–6.
3. Zhang, J., Ding, W. (2017). Prediction of Air Pollutants Concentration Based on an Extreme Learning Machine: The Case of Hong Kong. International Journal of Environmental Research and Public Health, 14 (2), 114. doi: https://doi.org/10.3390/ijerph14020114
4. Katranzhy, L., Podskrebko, O., Krasko, V. (2018). Modelling the dynamics of the adequacy of bank's regulatory capital. Baltic Journal of Economic Studies, 4 (1), 188–194. doi: https://doi.org/10.30525/2256-0742/2018-4-1-188-194
5. Manea, E., Di Carlo, D., Depellegrin, D., Agardy, T., Gissi, E. (2019). Multidimensional assessment of supporting ecosystem services for marine spatial planning of the Adriatic Sea. Ecological Indicators, 101, 821–837. doi: https://doi.org/10.1016/j.ecolind.2018.12.017
6. Çavdar, A. B., Ferhatosmanoğlu, N. (2018). Airline customer lifetime value estimation using data analytics supported by social network information. Journal of Air Transport Management, 67, 19–33. doi: https://doi.org/10.1016/j.jairtraman.2017.10.007
7. Kachayeva, G. I., Mustafayev, A. G. (2018). The use of neural networks for the automatic analysis of electrocardiograms in diagnosis of cardiovascular diseases. Herald of Dagestan State Technical University. Technical Sciences, 45 (2), 114–124. doi: https://doi.org/10.21822/2073-6185-2018-45-2-114-124
8. Zhdanov, V. V. (2016). Experimental method to predict avalanches based on neural networks. Ice and Snow, 56 (4), 502–510. doi: https://doi.org/10.15356/2076-6734-2016-4-502-510
9. Kanev, A., Nasteka, A., Bessonova, C., Nevmerzhitsky, D., Silaev, A., Efremov, A., Nikiforova, K. (2017). Anomaly detection in wireless sensor network of the «smart home» system. 2017 20th Conference of Open Innovations Association (FRUCT). doi: https://doi.org/10.23919/fruct.2017.8071301
10. Sreeshakthy, M., Preethi, J. (2016). Classification of human emotion from deap EEG signal using hybrid improved neural networks with Cuckoo search. Brain: Broad Research in Artificial Intelligence and Neuroscience, 6 (3-4), 60–73. Available at: https://www.slideshare.net/bpatrut/classification-of-human-emotion-from-deap-eeg-signal-using-hybrid-improved-neural-networks-with-cuckoo-search
11. Chica, J., Zaputt, S., Encalada, J., Salamea, C., Montalvo, M. (2019). Objective assessment of skin repigmentation using a multilayer perceptron. Journal of Medical Signals & Sensors, 9 (2), 88. doi: https://doi.org/10.4103/jmss.jmss_52_18
12. Massel, L. V., Gerget, O. M., Massel, A. G., Mamedov, T. G. (2019). The Use of Machine Learning in Situational Management in Relation to the Tasks of the Power Industry. EPJ Web of Conferences, 217, 01010. doi: https://doi.org/10.1051/epjconf/201921701010
13. Abaci, K., Yamacli, V. (2019). Hybrid Artificial Neural Network by Using Differential Search Algorithm for Solving Power Flow Problem. Advances in Electrical and Computer Engineering, 19 (4), 57–64. doi: https://doi.org/10.4316/aece.2019.04007
14. Mishchuk, O. S., Vitynskyi, P. B. (2018). Neural Network with Combined Approximation of the Surface of the Response. Research Bulletin of the National Technical University of Ukraine «Kyiv Politechnic Institute», 2, 18–24. doi: https://doi.org/10.20535/1810-0546.2018.2.129022
15. Kazemi, M., Faezirad, M. (2018). Efficiency estimation using nonlinear influences of time lags in DEA Using Artificial Neural Networks. Industrial Management Journal, 10 (1), 17–34. doi: http://doi.org/10.22059/imj.2018.129192.1006898
16. Parapuram, G., Mokhtari, M., Ben Hmida, J. (2018). An Artificially Intelligent Technique to Generate Synthetic Geomechanical Well Logs for the Bakken Formation. Energies, 11 (3), 680. doi: https://doi.org/10.3390/en11030680
17. Prokoptsev, N. G., Alekseenko, A. E., Kholodov, Y. A. (2018). Traffic flow speed prediction on transportation graph with convolutional neural networks. Computer Research and Modeling, 10 (3), 359–367. doi: https://doi.org/10.20537/2076-7633-2018-10-3-359-367

18. Bodyanskiy, Y., Pliss, I., Vynokurova, O. (2013). Flexible Neo-fuzzy Neuron and Neuro-fuzzy Network for Monitoring Time Series Properties. Information Technology and Management Science, 16 (1). doi: https://doi.org/10.2478/itms-2013-0007
19. Bodyanskiy, Ye., Pliss, I., Vynokurova, O. (2013). Flexible wavelet-neuro-fuzzy neuron in dynamic data mining tasks. Oil and Gas Power Engineering, 2 (20), 158–162.
20. Haykin, S. (1999). Neural Networks: A Comprehensive Foundation. Upper Saddle River, N.J.: Prentice Hall, Inc., 842.
21. Nelles, O. (2001). Nonlinear System Identification. Springer, 785. doi: https://doi.org/10.1007/978-3-662-04323-3
22. Wang, L.-X., Mendel, J. M. (1992). Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. IEEE Transactions on Neural Networks, 3 (5), 807–814. doi: https://doi.org/10.1109/72.159070
23. Kohonen, T. (1995). Self-Organizing Maps. Springer, 364. doi: https://doi.org/10.1007/978-3-642-97610-0
24. Kasabov, N. (2003). Evolving Connectionist Systems. Springer, 307. doi: https://doi.org/10.1007/978-1-4471-3740-5
25. Sugeno, M., Kang, G. T. (1988). Structure identification of fuzzy model. Fuzzy Sets and Systems, 28 (1), 15–33. doi: https://doi.org/10.1016/0165-0114(88)90113-3
26. Ljung, L. (1999). System Identification: Theory for the User. PTR Prentice Hall, Upper Saddle River, 609. Available at: https://www.twirpx.com/file/277211/
27. Otto, P., Bodyanskiy, Y., Kolodyazhniy, V. (2003). A new learning algorithm for a forecasting neuro-fuzzy network. Integrated Computer-Aided Engineering, 10 (4), 399–409. doi: https://doi.org/10.3233/ica-2003-10409
28. Narendra, K. S., Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. IEEE Transactions on Neural Networks, 1 (1), 4–27. doi: https://doi.org/10.1109/72.80202
29. Petruk, S., Zhyvotovskyi, R., Shyshatskyi, A. (2018). Mathematical Model of MIMO. 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T). doi: https://doi.org/10.1109/infocommst.2018.8632163
30. Zhyvotovskyi, R., Shyshatskyi, A., Petruk, S. (2017). Structural-semantic model of communication channel. 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T). doi: https://doi.org/10.1109/infocommst.2017.8246454
31. Alieinykov, I., Thamer, K. A., Zhuravskyi, Y., Sova, O., Smirnova, N., Zhyvotovskyi, R. et. al. (2019). Development of a method of fuzzy evaluation of information and analytical support of strategic management. Eastern-European Journal of Enterprise Technologies, 6 (2 (102)), 16–27. doi: https://doi.org/10.15587/1729-4061.2019.184394
32. Koshlan, A., Salnikova, O., Chekhovska, M., Zhyvotovskyi, R., Prokopenko, Y., Hurskyi, T. et. al. (2019). Development of an algorithm for complex processing of geospatial data in the special-purpose geoinformation system in conditions of diversity and uncertainty of data. Eastern-European Journal of Enterprise Technologies, 5 (9 (101)), 35–45. doi: https://doi.org/10.15587/1729-4061.2019.180197